REMARKS

Favorable reconsideration of the present application, in view of the present amendment and in light of the following remarks, is respectfully requested. Claims 1-19 are currently pending in this application. No claims have been added or canceled herewith. Claims 1, 8 and 15 have been amended herewith to recite that the call flow event queues are dedicated to their corresponding threads, although it is believed that one of ordinary skill in the art would have already understood this from the previous claims. Such a change is supported by the previously pending claims as well as by Figures 4A and 4B and their corresponding description starting at page 16, line 8. Thus, no new matter has been added.

In the outstanding office action, claims 1-19 were rejected under 35 USC 103(a) as being obvious over a three-reference combination including U.S. Patent Nos. 5,549,864, 6,289,369 and 6,909,708 (hereinafter "the '864 patent", "the '369 patent", and "the '708 patent", respectively). Applicant respectfully traverses the rejection (1) as not disclosing all of the elements of the claims and (2) as lacking a permissible motivation to combine the references.

Claim 1 recites "a method ... of distributing call flow events among a plurality of threads, each thread having an associated call flow event queue in which call flow events are queued" and "reassigning a call flow event from the call flow event queue dedicated to the first thread to the call flow event queue dedicated to a second of the plurality of threads." Such positively recited limitations are not taught by the combination of references.

The Office Action alleges that the '369 patent "teaches the invention as claimed, including associating threads and queues on a one-to-one basis (col. 8 lines 40-64) such that distribution of events among a plurality of queues amounts to distributing events among a plurality of threads (col. 5 lines 59-67)." This assertion, however, ignores the recitation of "call flow event queues" and attempts to utilize the scheduling queues of the '369 patent in their place. This assertion is also incorrect in that the threads and queues are not associated in a one-to-one basis but rather on a many-thread-to-one-queue basis, as described in greater detail below.

Col. 8 lines 40-64 of the '369 patent are part of a larger discussion entitled "Locality and Load Balancing" beginning at col. 8 line 14. The '369 patent discloses three types of threads: "(a) non-sticky (schedulable on any of the available processors 10); (b) part-sticky (schedulable on a set or subset of the available processors 10); or (c) sticky (schedulable only on a specified processor 10 and not migratable at all)." How those threads are scheduled is then discussed starting at col. 8, line 24 which states "Non-sticky and part-sticky threads 22 are scheduled using the central schedule queue 26, which ensures load balancing. The use of the central schedule queue 26 implies that a thread 22 that can migrate among processors 10. However, non-sticky and part-sticky threads 22 are more expensive to schedule and context-switch than sticky threads 22 that are scheduled using the per-processor local schedule queues 28." (Emphasis added.) Thus, the scheduling queues are "containers" that plural threads are put into until they can be scheduled (i.e., allowed to execute on a processor resource). In fact, the very section cited by the office action, starting at col. 8, lines 52, states "Appropriate scheduling is done by clustering the threads 22 with the same affinity-id 36 together in a per-processor local schedule queue 28. ... If there is no eligible thread 22 in the per-processor local schedule queue 28, then a thread 22 from the central schedule queue 26 is dispatched for execution." (Emphasis added.)

Furthermore, in the second section cited by the office action, col. 5 lines 59-67, the '369 patent re-emphasizes that threads are being put into scheduling queues. It states:

> FIG. 3 is a block diagram that illustrates an exemplary central schedule queue 26 and per-processor local schedule queues 28 implemented according to the preferred embodiment of the present invention. Threads 22 resident in the central schedule queue 26 may be scheduled for execution on any available processor 10 by the scheduler 24, while threads 22 resident in the per-processor local schedule queues 28 may be scheduled for execution only on the specified processor 10 (when available) by the scheduler 24.

(Emphasis added.) Thus, the "queues" of the '369 patent are not call flow event queues as claimed but are instead scheduling queues. Moreover, the office action later asserts that the '369 patent "expands on the one-to-one relationship of queues and processors to include 'sticky' threads, where each processor has a thread that services a dedicated queue." Office action, page 3, lines 17-19. However, as can be seen from the description of the utilization of the scheduling queues above, threads do not "service" a dedicated queue but rather are placed into a queue for scheduling. Moreover, "sticky" threads are placed in the same scheduling queue all the time, so they are actually the opposite of threads that can be distributed to balance load. See col. 8 lines 24-27 which states "Non-sticky and part-sticky threads 22 are scheduled using the central schedule queue 26, which ensures load balancing. The use of the central schedule queue 26 implies that a thread 22 that can migrate among processors 10."

Even assuming that the office action were to reverse its position and allege that the non-sticky and part-sticky threads were relevant to the present invention, this would still not be sufficient to render obvious the present invention. The non-sticky and part-sticky threads utilize a central scheduling queue that is also not the recited call flow event queues. Thus the non-sticky and part-sticky threads and their queue do not anticipate the same limitation not anticipated by the sticky threads discussed above.

The '369 patent also does not anticipate or render obvious the "reassigning a call flow event from the call flow event queue dedicated to the first thread to the call flow event queue dedicated to a second of the plurality of threads." The '369 patent utilizes the migration of non-sticky and part-sticky threads as the basis of its load balancing. However, the claims recite not moving the threads themselves but moving a call flow event from the call flow event queue dedicated to the first thread to the call flow event queue dedicated to a second of the plurality of threads. Thus, not only is the limitation not met, it utilizes an entirely different principle of operation than the '369 patent.

The office action further cites the '864 patent as rendering at least a portion of claim 1 obvious. The office action alleges that the '864 patent teaches "determining a workload level for each of the plurality of queues." However, claim 1 recites "determining a call flow workload level for each of the plurality of <u>threads</u>." Moreover, the '864 patent does not teach utilizing threads at all – let alone threads with dedicated call flow event queues. In the event that the office action is attempting to suggest that the system of the '864 patent is <u>inherently</u> multi-threaded or even thread-based, it is respectfully requested that the next office action identify on what basis or facts such an assertion of inherency is based. It is respectfully noted that operating systems need not inherently support multiple threads.

Since the '369 patent teaches utilizing queues other than call flow event queues, and since the queues are not provided in a one-to-one relationship, the combination of the '369 patent and the '864 patent does not teach or suggest the same positively recited limitations of: (1) distributing call flow events among a plurality of threads, each thread having a dedicated call flow event queue in which call flow events are queued and (2) "reassigning a call flow event from the call flow event queue dedicated to the first thread to the call flow event queue dedicated to a second of the plurality of threads." Thus the combination fails to teach the same positively recited limitation not taught by the references individually.

Lastly the office action cites the '708 patent as teaching the "application of load balancing techniques to internet telephony." However, nothing in the cited section describes that the load-balancing is performed using the claimed method of "reassigning a call flow event from the call flow event queue dedicated to the first thread to the call flow event queue dedicated to a second of the plurality of threads." In fact, the word "thread" does not appear in the '708 patent at all, although at least some of the operating systems disclosed therein are capable of utilizing threads.

It also appears that the gateways 1081, 1084 and 1086 are disjoint systems connected by an Ethernet such that it is unlikely that the "load-balancing" discussed would be moving call flow events from one call flow event queue to another. Instead, the load balancing could be other techniques, such as utilizing DNS aliasing or assigning which gateway to use based on a round-robin approach. See, e.g., U.S. Patent No. 6,888,836, filed herewith in an IDS. Thus, the mere reference to "load-balancing" does not mean that the claimed implementation is anticipated or rendered obvious. Accordingly, even the combination of all three references fails to teach the same positively recited limitations not taught by the references individually.

There is also no tenable motivation to combine the references in the fashion asserted in the office action. As discussed above, the motivation that the '369 patent "expands upon the one-to-one relationship of queues and processors to include 'sticky' threads, where each processor has a thread that services a dedicated queue" is incorrect since there are multiple-threads per scheduling queue. Moreover, even if the '708 patent did disclose that load balancing was important in internet telephony, there is no indication that it would have been performed in the claimed manner or that one of ordinary skill in the art would have been motivated to include it in the claimed manner. Further, there is no indication why one of ordinary skill in the art reading the '708 patent, directed to telephony systems, would have consulted the '864 patent, directed to Data Move Processors of an input/output subsystem. The fact that both patents utilize the phrase "load-balancing" is insufficient to assume a motivation to combine those particular references in light of the multitude of other references that also utilize the same phrase.

Claim 8 similarly recites "program code configured to reassign a call flow event from the call flow event queue dedicated to the first thread to the call flow event queue dedicated to a second of the plurality of threads." Thus, claim 8 and its dependent claims are patentable over the cited combination of references for reasons analogous to the reasons set forth above for the patentability of claim 1.

Claim 15 similarly recites "a call flow manager configured to distribute a plurality of call flow events among a plurality of threads used for managing the processing of a plurality of call flows, the call flow manager optimizing the processing of the call flows by determining which of the plurality of threads are operating inefficiently and reassigning a portion of the call flow events assigned to the dedicated call event queue of the inefficient thread to the dedicated call event queue of another of the plurality of threads having excess call flow processing capacity." Thus, claim 15 and its dependent claims are patentable over the cited combination of references for reasons analogous to the reasons set forth above for the patentability of claim 1.

Consequently, in view of the present amendment and in light of the above discussions, the outstanding grounds for rejection are believed to have been overcome and in condition for allowance. An early and favorable action to that effect is respectfully requested.

Respectfully submitted,

DAVIDSON, BERQUIST,
JACKSON & GOWDEY, L.L.P.

[illegible]

Michael R. Casey, Ph.D.
Registration No. 40,294

CUSTOMER NUMBER

42624

Davidson Berquist Jackson & Gowdey, LLP
4300 Wilson Boulevard, 7th Floor
Arlington, VA 22203
Ph: 703-894-6400
Fax: 703-894-6430